Subject: Re: [patch 5/9] unprivileged mounts: allow unprivileged bind mounts
Posted by Jan Engelhardt on Wed, 09 Jan 2008 12:44:31 GMT

View Forum Message <> Reply to Message

On Jan 8 2008 20:08, Miklos Szeredi wrote:

```
>> On Tue, 2008-01-08 at 12:35 +0100, Miklos Szeredi wrote:
>> > +static int reserve_user_mount(void)
>> > +{
>> > +      int err = 0;
>> > +
>> > +      spin_lock(&vfsmount_lock);
>> > +      if (nr_user_mounts >= max_user_mounts && !capable(CAP_SYS_ADMIN))
>> > +              err = -EPERM;
>> > +      else
>> > +              nr_user_mounts++;
>> > +      spin_unlock(&vfsmount_lock);
>> > +      return err;
>> > +}
>>
>> Would -ENOSPC or -ENOMEM be a more descriptive error here?
>
>The logic behind EPERM, is that this failure is only for unprivileged
>callers.  ENOMEM is too specifically about OOM.  It could be changed
>to ENOSPC, ENFILE, EMFILE, or it could remain EPERM.  What do others
>think?
```

ENOSPC: No space remaining on device => 'wth'.
ENOMEM: I usually think of a userspace OOM (e.g. malloc'ed out all of your 32-bit address space on 32-bit processes)
EMFILE: "Too many open files"
ENFILE: "Too many open files in system".

ENFILE seems like a temporary winner among these four.

Back in the old days, when the number of mounts was limited in Linux, what error value did it return? That one could be used.

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers